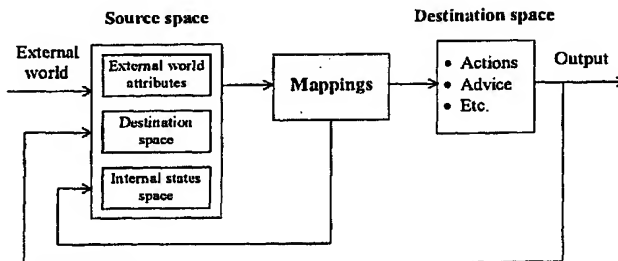




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 17/00		A1	(11) International Publication Number: WO 99/66420
			(43) International Publication Date: 23 December 1999 (23.12.99)
(21) International Application Number: PCT/AU99/00501 (22) International Filing Date: 18 June 1999 (18.06.99) (30) Priority Data: PP 4164 18 June 1998 (18.06.98) AU (71) Applicant (for all designated States except US): SAVANT SYSTEMS INTERNATIONAL PTY. LTD. [AU/AU]; Level 21, 201 Miller Street, North Sydney, NSW 2060 (AU). (72) Inventor; and (75) Inventor/Applicant (for US only): GUIGNARD, Paul [AU/AU]; Level 21, 201 Miller Street, North Sydney, NSW 2060 (AU). (74) Agent: F.B. RICE & CO.; 605 Darling Street, Balmain, NSW 2041 (AU).		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published With international search report.	

(54) Title: GENERIC KNOWLEDGE MANAGEMENT SYSTEM



(57) Abstract

A computerized generic knowledge management system, comprising a multidimensional global space defined by attributes which are dimensions in the global space. Each attribute defines a feature of the external world or the internal state of the system, or actions that can be taken to modify them. There is a source space, within the global space, made up of selected attributes to define a context in which to state problems. There is a destination space, within the global space, made up of selected attributes to define a context in which to provide answers to problems stated in the source space. Mapping occurs between the source space, representing stated problems, and the destination space, representing answers expressing and embodying knowledge supplied by experts appropriate to the respective problems stated in the part of the source space.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Generic Knowledge Management System

Technical Field

5 This invention concerns a computer technology for the design of knowledge systems, for the acquisition, modelling, storage and access of knowledge for these systems, and for the maintenance of that knowledge. This technology will be referred to as Generic Knowledge Management System (GKMS).

10 Background Art

Knowledge-based systems (KBSs) are a class of programs in which the knowledge about a range of problems and their solutions is stored separately from the code that is used to manipulate that knowledge to produce solutions. KBS design is different from traditional software where the
15 knowledge about a problem and its solution is integrated into the code that processes that knowledge. Because of the separation between knowledge and its processing, KBSs, in principle, should be easier to develop and maintain and safer to use than traditional programs. However it is not the case in practice and the penetration of KBSs in industry is very small compared to
20 spreadsheets, databases and word processors

Summary of the Invention

In a first aspect the invention is a computerised generic knowledge management system, comprising:

25 a multi-dimensional global space within computer memory defined by attributes, where each attribute defines a feature of the external world or the internal state of the system, or actions that can be taken to modify them, and each attribute is a dimension of the global space;

30 a source space, within the global space, made up of selected ones of the attributes to define a context, in which to state problems;

a destination space, within the global space, made of selected ones of the attributes to define a context in which to provide answers to problems stated in the source space;

35 mappings between defined parts of the source space which each represent one or more stated problems, to defined parts of the destination space which each represent one or more answers expressing and embodying

knowledge supplied by experts appropriate to the respective problems stated in the part of the source space.

The system may enjoy a number of advantages. For instance, The system is able to inform a user when the answer to a problem falls outside its knowledge, that is the problem stated is outside a defined part of the source space. In this case the system is able to ask a domain expert (that is an expert in that domain of knowledge) to increase its knowledge by providing the answer and defining an appropriate part of the source and destination space and an appropriate mapping. The experts are also able to maintain the system's knowledge by correcting existing defined parts. In this way the system is able to get more knowledgeable as it is being used.

Knowledge acquisition may be incremental and interactive. In addition the knowledge is certified, that is verified and validated, by the expert at acquisition time.

Since the system knows the limits of its knowledge it is able to restrict the use of its knowledge to situation it recognises, that is to problems that fall within the regions in the source space. These regions have been approved by domain experts.

Such systems can form natural extensions of human beings for the management of their knowledge. It is possible for humans to use the system to store knowledge, retrieve it, modify it, extend it and share it with other experts and with users who are not experts. Domain experts can use such systems as repositories of their knowledge and as tools for the development and codification of new knowledge acquired as they gain experience in their domain of expertise.

The defined parts of the source and destination spaces may be points or regions. The destination space may also be part of the source space. The two spaces can overlap.

The mapping process can be explanations or actions. Explanation mappings are not calculated, they are stated. Explanation mappings are associated with code which enable the outcome to be displayed on a screen or printer. Action mappings may associate a situation in the source space to actions expressed in the destination space. In this case the destination space is made of instructions to be carried out by agents. Alternatively, action mappings may be specified by a function or module that can be calculated, using the values of the source attributes that define the situation as

parameters. The result of these mappings are attributes which can take values.

Source and destination space editing sub-systems may enable authorised users to define and modify the destination and source spaces or contexts.

A mapping editing sub-system may enable experts to define mappings which embody knowledge. This sub-module may present the expert with the source context which allows the selection of attributes which belong to a region and the specification, for each attribute, of the range of values which mark the borders of the region in each dimension, each attribute is a dimension. The region specifies the conditions which determine whether a mapping can fire. The process is similar for the destination space. The two regions are then linked and identified as a mapping.

The defined parts of the source context determine when a mapping can fire. This process is 'location independent'; that is, it is independent from where the mapping is located in a system. Location independence is seen as a major advantage in that it frees developers from the issue of location. Processing behaviour depends only on the conditions for a mapping to take place, expressed as a defined part in the source space. There is however an additional load put upon the system that now has to find which mapping applies next, a requirement that is taken care of in normal programming by the location of the code.

A composite system may comprise a collection of systems in which the source contexts of the systems are united, the destination contexts of the systems are united and the mappings of the systems are united to form the composite.

A composite system made of systems with overlapping contexts is referred to a knowledge base. Composite systems can be concatenated or grouped to build larger knowledge bases.

A query definition sub-system may enable a user to define a query in the source space, or a process to perform the role of a user. In effect, the user defines a situation for the knowledge processing module to act on. Each object or attribute in the source space can take three values in an inquiry: 'specified', 'don't know' and 'unspecified'.

The query definition can be non-interactive, in which case the system presents the user with the complete source space. The user then specifies a situation and the system looks for regions compatible with the query. If there are compatible regions, then their outcomes are presented to the user. If there isn't any compatible region, the user is informed.

Alternatively the query definition may be interactive, in which case the system presents the user with one or a few questions at a time for the user to answer. Once it is done, the system processes the answers and determines the next best questions to ask. The best questions are those that lead to the mappings, that is regions, compatible with the query with as few questions as possible. When the system has identified a compatible region, then it presents its outcome to the user. If the system cannot find a compatible region, it informs the user.

A hybrid query definition is also possible where the system presents the user with several questions grouped logically, that is addressing a single issue. Once the user has answered these questions, the system presents the next group of questions, or single question, as the case may be.

Knowledge processing may be used to identify whether there are regions that are compatible with the inquiry. Knowledge processing starts with a list of candidate regions or mappings, initially all the regions in the knowledge base, and on the basis of the objects specified in the inquiry, that is the values of the dimensions in the source space, determines: the regions that are ruled out by the inquiry; the regions that are compatible with the inquiry; and the regions that are undetermined, that is, the regions for which one cannot say whether they are compatible or ruled out because some questions have not yet been asked.

The process may stop when: there are no more candidate regions or mappings; there are no more undetermined regions or mappings; or there are no more objects whose values have not been determined.

In non-interactive processing the system takes as input the query defined by the user and looks for regions in the source space that are compatible with the query. A region is compatible with the query if the region contains (as defined) the query. This means that the value of each object in the region contains the value of the object in the query

In interactive processing the system calculates the discriminating power (as defined) of each object, or dimension, in the source space the value

of which has not yet been specified. Once the discriminating power of each object is calculated, the system asks the questions with the highest discriminating power. If several objects have the same discriminating power, then the system either presents all the related questions or selects one of them only for presentation to the user. Once the user has supplied the answer(s) related to the object(s) with the highest discriminating power, the system uses the answer(s) to remove from the list of candidate all the knowledge items that are ruled out by the answer. The process described above is then repeated.

Interactive processing can be either forward chaining or backward chaining.

In another aspect the invention is a data acquisition method for a computerised generic knowledge management system, comprising the steps of:

inspecting a problem that either has no answer or an answer which is deemed to be inadequate, that is a problem for which there is no defined part of the source space;

specifying attributes, and if appropriate, explanations relevant to the problem;

defining the solution to the problem;

generalising the source context to generalise the inquiry to a larger part of the source space; and

saving the knowledge item generated.

The solution may be defined after the source context has been generalised.

An inquiry may be accompanied by a message from the user who filed the inquiry. The message may describe the inquiry more specifically or in more details.

The context, of either the source or destination may be reduced or enlarged. New attributes can be added.

Knowledge items or mappings need to be ordered according to their fit with the enquiries. This applies to definite outcomes only or to definites and candidates outcomes. The fit is determined by three factors:

1. Weight of each source attribute, weights may not all be equal.

2. For definite and candidate knowledge items: proportion of attributes in the inquiry which are satisfied by the region attributes; that is attributes belonging to the region of the knowledge item.

3. For definite and candidate knowledge items: proportion of the attributes
5 in the source context which belong to the region of a candidate knowledge item.

Domain experts need to manage the knowledge base from the point of view of knowledge comprehensiveness, consistency, quality, etc. The tools available to assist them are to:

10 Detect overlapping knowledge items, in their source and/or destination to ensure they are compatible.

Identify knowledge items with a specified combination of source and destination attributes or values

Edit existing knowledge items.

15 Inspect history of knowledge items in the system:

- list of knowledge items (active and deactivated)
- when created, by whom.

Brief Description of the Drawings

20 Examples of the invention will now be described with reference to the accompanying drawings, in which:

Figure 1 is a diagram showing knowledge in the cognitive model.

Figure 2 is a cognitive model reasoning flow chart.

25 Figure 3 illustrates the mapping from source space to destination space.

Figure 4 is a block diagram illustrating the composite GKMS module design.

Figure 5 is a block diagram showing the network of GKMS knowledge bases.

30 Figure 6 is a flowchart showing knowledge processing.

Figure 7 is a diagram of the elementary GKMS model architecture.

Figure 8 illustrates the definite and candidate knowledge items.

Figure 9 illustrates the definite and candidate knowledge items.

35 Figure 10 illustrates the architecture for a computer system based on the cognitive model.

Figure 11 is a block diagram of the architecture for a user only system based on the cognitive model.

Best Modes of the Invention

5 The Cognitive Model

The invention is based on a cognitive model in which expertise is applied locally and then an attempt is made to generalise. That is, a domain expert first determines the attributes necessary to define an explicit global context for a class of problems and solutions. The domain expert then
10 specifies problem and solution contexts within the global context using the attributes. The domain expert then defines a specific problem within the problem context using the attributes. The domain expert next provides a specific answer to the specific problem. Finally the domain expert expands the specific problem into a region of the problem context where the same
15 specific answer, taking account of its context, is deemed by the expert to hold.

Once a region of a problem and an answer have been produced by a domain expert, this knowledge can be used by non-experts to find the answer to problems that fall within the region.

20 A useable system requires the definition of several, sometimes many, regions and associated answers. These regions are typically defined when a problem is encountered for which there is no existing answer in the system, that is, the problem does not fall within an existing region. Alternatively, the domain expert can decide to define regions in the problem space based on
25 hypothetical problems.

A knowledge item consists of a region of a problem and its associated answer, both defined in their respective contexts, and it may be expressed as a mapping. The model assumes that knowledge has no global validity, but that knowledge is inseparable from its context.

30 This is represented in Figure 1 where the global context is a space with attributes along the axes. In Figure 1 there are six regions in a 2-Dimensional problem plane. The precise problems that triggered the definition of the regions are shown as • in the plane, and six corresponding answers are shown as • on the "answers" axis.

35 Notable features are:

1. The regions can be of different shapes and sizes, as judged appropriate by the domain expert.

2. The precise problem does not need to be in the centre of the region. Again this is judged by the domain expert.

5 3. Some regions can be defined without being prompted by a precise problem. In this case there is no • in the region (region r4 in Figure 1).

4. Separate regions can have the same answer. This could mean that the domain expert took a conservative approach at the time of the region definition and certified the answer only in a small region. Later, when
10 defining another region, perhaps prompted by a different precise problem, the expert gave the same answer but again felt confident only with respect to a small local region.

5. Regions can overlap. In this case a strategy must be developed to select the appropriate answer among the several that may be possible.

15 Although the answer space is represented as a single axis in Figure 1, it usually is multidimensional, with each outcome being a partition of the total solution space.

Knowledge Acquisition

20 The process followed by a system based on the model for knowledge acquisition is illustrated in Figure 2.

The process steps are:

1. The system invites a user to define a specific problem.

2. The system checks whether the problem falls within a region defined
25 in the problem context.

3. If it does, the system gets the answer corresponding to that region and gives it as answer to the new problem.

4. If the problem does not fit inside a defined region, then the system informs the user that no answer can be given at this stage.

30 5. The system then refers the problem to a domain expert who:

a) specifies the answer to the problem, and

b) defines a region or mapping for that answer.

6. The system then links the region with the answer and adds both to its knowledge base. This new knowledge now becomes available for all
35 subsequent users of the system.

With respect to point 5 above, the domain expert vouches for the region and its answer. In effect this verifies the answer is the appropriate one for the problem, and validates the answer is relevant in the context.

5 Knowledge Representation Features

Knowledge is local in that each knowledge item is a problem region and its answer in a global context which determines the types and ranges of problems that can be addressed. It is the role of domain experts to use their experience and judgment to define the global context. Because of the local
10 property of knowledge in the model, it is possible to define a global context with a different number of attributes, or dimensions, for different knowledge items, and for the problem regions and the answers.

Knowledge Acquisition Features

15 With reference to Figure 2, the key features are:

1. Knowledge acquisition is incremental, usually prompted by an enquiry about a specific problem.
2. A system is useable even when little knowledge is included in it. As the system is used, more knowledge is entered and the usefulness grows.
- 20 3. Only useful knowledge, that is knowledge relevant to specific problems, is acquired. (Since domain experts can define regions and outcomes without being prompted by enquiries, it is possible to include non-useful knowledge in the system, that is knowledge that will not be of use in dealing with practical problems. This is however unlikely to happen to any
25 significant extent in practice.)
4. Knowledge is verified at acquisition time. That is, the domain expert vouches for the applicability and relevance of the knowledge to all situations and problems that fall within the region. This is because the definition of most of the knowledge items stored in the system is triggered by specific
30 problems. The expert can be either very conservative and define a small region or more confident and define a larger region. When regions and outcomes are defined without being prompted by a specific problem, then the regions defined may have little relevance to the intended domain of applications for the system and one cannot claim that these knowledge items
35 are validated.

5 5. The knowledge items have independent validity. An item can be removed without affecting the validity of the other items of the system. The rest of the system is still useable. A knowledge item can be removed and replaced by another. In a similar way, a region or its outcome can be modified without affecting the validity and the operations of the rest of the system. The system is therefore robust. In contrast traditional rule-based systems are not robust in that they can be drastically affected by any changes in the rule base.

10 6. The knowledge items may be independent of each other. However, where new knowledge items are defined from new enquiries that do not fall within existing regions they may be dependent. The definition and storage of new knowledge items is in part determined by the knowledge already existing in the system.

15 The Reasoning Process consists of:

1. Knowledge acquisition support

20 When a new region and answer is being defined by a domain expert, it can happen that this region overlaps with previously defined regions. Support is required to let the domain expert know that these overlapping regions exist and to assist them in viewing them. The domain expert then can opt to:

- a. modify the new region or its outcome, or both, appropriately.
- b. modify one or several already defined regions or their outcomes, or both, provided the domain expert has the authorisation to do so.

25 It is not necessary to remove all overlaps between regions. A case contained by several regions may have several possible or complementary answers, one from each region containing it. It is also possible that two regions have the same outcome. In this situation, the answer is the same whatever region one selects.

30 Knowledge acquisition support is not mandatory. It is possible, for example, to simply give as answer to an enquiry at run time the outcome of the most recent region containing the enquiry.

2. Solution Search Processing, consisting of:

35 a. Finding the regions that contain the enquiry. This can be done in two ways:

- A non-interactive search:

Selecting the region or regions most relevant to the enquiry
Retrieving the outcome or outcomes corresponding to these regions.
Presenting these outcome in the appropriate way to the user.

- An interactive search:

- 5 defining the enquiry as completely as possible
 finding the regions that contain it.

Interactive searching then involves asking the user only these
questions about the problem that enable the system to arrive at a solution as
quickly as possible. Each question elicits some features about the enquiry
10 and directs the search towards the most relevant part of the problem space.
Some features of an enquiry may not be asked as they apply to regions in a
part of the problem space that has already been found as being irrelevant to
the enquiry. Interactive searching can be achieved, for example, by
measuring the "regions discriminating power" of each feature in the problem
15 space and by asking, at each step in the inferencing process, only about these
features that have the highest discriminating power.

The Elementary GKMS Module

20 The elementary GKMS module is the basic building block for all
GKMS implementations. It comprises a source context or space, a destination
context or space, and a mapping that can be action or explanation (that is, at
least one knowledge item).

25 The source space is made of attributes or objects that enable the expert
user to define situations. The source space typically contains attributes that
define the external world. However it can also contain attributes that define
the internal state of the system. When it is the case, mappings can be used to
take actions based on the state of the system, that is, mappings can be used
to control the behaviour of the system.

30 The destination space is made of attributes that describe actions,
events, statements about the external world. It specifies the actions that can
be taken to modify the external world or the internal state of the system.

The destination space can also be part of the source space. The two
spaces can overlap.

35 The mappings express and embody the knowledge supplied by experts
to provide answers, expressed in the destination space, appropriate to the
situations described in the source space. A mapping is a relationship

between part of a source space onto part of a destination space, typically from a situation or group of situations in the source space onto an outcome in the destination space. In Figure 3, all the situations in the source space which are part of the region are linked to the same outcome. Both spaces are multi-dimensional.

The curved arrow, with its attendant source and destination spaces, its region from the source space to its outcome in the destination space represent the mapping. It specifies a region in the problem space and a way to produce an outcome when the conditions of the problem place it within the region. The mapping process can be explanations or actions. The source and destination spaces define the context for the mapping..

Explanation Mapping

Explanation mappings are defined by their source and their destination. They are not calculated, they are stated. Explanation mappings are associated with code which enable the outcome to be displayed on a screen or printer.

Action Mapping

Action mappings come in two types:

- Type 1 mappings associate a situation in the source space to actions expressed in the destination space. The destination space, instead of being explanations as in explanation mappings is made of instructions to be carried out by the some agents.

- Type 2 mappings are specified by a function or module that can be calculated, using the values of the source attributes that define the situation as parameters. The result of type 2 mappings are attributes which can take values. Programming can be viewed as the calculation of action mappings, one after another.

Action mapping examples:

- A workflow module in which the action to be taken by the system, outcome, are predicated by a situation described in the source space (type 1).
- An equation with variables and constants that are part of the source and destination spaces (type 2).
- A procedure or function (algorithm) (type 2).

The regions in the source space can be small or large. Regions can overlap and have sub-regions. The outcome can be a point or a region in the destination space. The mapping process described above is a generic process that covers all that can be expressed using logical and mathematical expressions.

Time can be an attribute, or object, both in the source and destination spaces.

The elementary GKMS module supports the following functions:

- A source and destination spaces editing sub-module
- A explanation mapping editing sub-module
- An action mapping editing sub-module
- A knowledge base
- A query definition sub-module
- A processing sub-module
- A system behaviour sub-module

The GKMS module has two types of users:

The expert or experts who are responsible for defining the source and destination spaces, and for entering and managing the mappings, or knowledge, in the module. They are accountable for the accuracy and currency of the mappings, or knowledge.

The non-experts, or users, who defines enquiries interactively with the system, with a view of getting knowledge, such as advice and recommendations, from the system. These are the results of actions carried out by the system.

Source And Destination Context Editing Sub-Module

This sub-module enables authorised users to define and modify the destination and source spaces or contexts. These spaces are constructed out of attributes or objects. Contexts are defined as a hierarchy of attributes which are categorised in groups of folders. This is similar to the file system on a personal computer, with enables users to organise their files or documents in folders and subfolders. Context editing enables experts to define attributes or objects used to describe the conditions for a mapping to take place (source context) and the range of possible outcomes of mappings

(destination context). It also enables the experts to organise these attributes in folders and subfolders. The source and destination contexts represent the domain in which mappings (or knowledge) are expressed. As such experts need to include in the source and destination contexts all the relevant attributes which define the domain of applicability of the mappings.

5 Table 1 shows these objects with their properties and types. The dimensions of the source and destination spaces are equal to the number of objects or attributes in these spaces.

Table 1: Context objects and their properties

Description	Properties
<ul style="list-style-type: none">· Can act as a folder and/or attribute· As folder: container for other objects, identifies a class· As attribute: specifies part of a condition, an explanation or an action	<ul style="list-style-type: none">· Object number· Title· Object type (folder or attribute)· Has members (objects)· Some or none of its objects specified· Multimedia explanation (each attribute can be explained by the expert)· Date created· Author· Status: active or deactivated (a deactivated attribute is one that is no longer part of the current context and cannot be used)· Mappings it belongs to· When attached to a mapping:<ul style="list-style-type: none">- complete source and destination contexts available when mapping was defined- importance level in determining whether the mapping can fire- confidence level that the attribute belongs to the outcome· Attribute type (explanation or action)· Explanation: list, logical, numeric, text, constant and their allowed values or ranges of values· Relationships between values: any, all, not· Action: software, its inputs and outputs

Table 2: Context object definition

Properties (fields)	Explanation
Title	
Explanation	· Multimedia description of the knowledge or action embodied in the mapping
Number	· Unique number
Date created	
Author	
Status (active or inactive)	· A deactivated mapping is one that is still in the elementary GKMS module but that cannot fire
Knowledge items it belongs to: - to regions for source attributes - to outcomes for destination attributes	· Knowledge items identified by their numbers or titles

The implementation of this sub-module can be done easily using modern software development tools.

5 Mapping Editing

Mapping editing enables experts to define mappings which embody knowledge. Mapping definition takes place in the source and destination contexts.

Experts define a region in the source space, and attach it to an outcome in the destination space, the outcome can also be a region. This sub-module presents the expert with the source context which allows the selection of attributes which belong to a region and the specification, for each attribute, of the range of values which mark the borders of the region in each dimension, each attribute is a dimension. The region specifies the conditions which determine whether a mapping can fire. The process is similar for the destination space. The two regions are then linked and identified as a mapping. Each item is an object in the GKMS module.

The two contexts, source and destination, are explicit and, jointly, form the domain of discourse or expertise for the mapping. Mappings represent knowledge with respect to their explicit domain of discourse. This point is very important as any form of knowledge is context dependent; that is, it is associated to an explicit domain of discourse. When a mapping is

dissociated from its domain of discourse (by expressing only its region and outcome without specifying the complete source and destination contexts for example) then it ceases to represent precise and reliably useful knowledge. As source and destination contexts can vary, a module can contain mappings each defined with respect to a different domain of discourse.

The elementary GKMS module enables experts to define knowledge in the form of explanation or action mappings, which is explicitly context dependent.

The region in the source context determines when a mapping can fire. This process is 'location independent'; that is, it is independent from where the mapping is located in a system. This can be contrasted to usual programs where the knowledge about the processing is located in the code itself and the operations depend on the location of the code in the program. Location independence is seen as a major advantage in that it frees developers from the issue of location. Processing behaviour depends only on the conditions for a mapping to take place, expressed as a region in the source space. There is however an additional load put upon the system that now has to find which mapping applies next, a requirement that is taken care of in normal programming by the location of the code.

Explanation Mapping Editing

This sub-module enables experts to define and edit mappings, or knowledge items, of the explanation type.

Table 3: Object range specification for source and destination attributes in explanation mappings

Space attribute	Source and destination spaces range specification
List variable	<ul style="list-style-type: none"> Select more than one item in its list as compatible items All items except those selected
Logic variable	<ul style="list-style-type: none"> Specify 'true' or 'false' Specify 'does not matter'
Numeric variable	<ul style="list-style-type: none"> Specify compatible range, or Specify complement of compatible range Specify 'does not matter'

Date/time variable	<ul style="list-style-type: none"> Specify compatible date or range of dates Specify complement of compatible date or range of dates Specify 'does not matter'
Text variable	<ul style="list-style-type: none"> Specify exact or partial compatible match, or Specify complement of exact or partial compatible match Specify 'does not matter'

5 The specification 'does not matter' can be either explicit or implicit. In the first case, the expert has to specify the range (even 'does not matter') for each object in the region in the source space. In the second case, the objects are assumed to have the default value 'does not matter' unless a range or value is specified.

10 When an object in the destination space is attached to a region, that is the object becomes part of the outcome of an inference process, then it automatically becomes a member of the source space, preferably into a folder labelled 'inferred objects'. An 'inferred object' can also be identified with an icon or a colour and be located anywhere in the source space list. When an inferred object is attached to a region in the source space, then the source and destination spaces are said to be overlapping. When an inferred object is attached to a region, the region is then labelled as 'inferred region' (an
15 inferred region has at least one inferred object). Objects and regions that are not inferred are also described as primary objects and regions.

Figure 1 illustrates explanation items in the source-destination context. The • in all regions except r4 indicate that an enquiry was presented for which there was no answer. The system then presented the enquiry to an expert who attached it to an outcome. The expert defined a region around the
20 enquiry so that all future enquiries falling inside the region produce the same outcome. This last step add usefulness to the knowledge stored in the System as the item becomes applicable to a range of situations rather than to one situation only.

25 The explanation mapping is specified by the properties listed in Table 4.

Table 4: Explanation mapping definition

Properties (fields)	Explanation
Title	

Summary	<ul style="list-style-type: none"> • Multimedia description of the knowledge or action embodied in the mapping
Number	<ul style="list-style-type: none"> • Unique number
Date created	
Author	
Status (active or inactive)	<ul style="list-style-type: none"> • A deactivated mapping is one that is still in the elementary GKMS module but that cannot fire
Attributes in the source context	<ul style="list-style-type: none"> • List of attributes (identified by their numbers or titles)
Attributes in the destination context	<ul style="list-style-type: none"> • List of attributes (identified by their numbers or titles)
Source attributes in the region and their ranges	<ul style="list-style-type: none"> • Specifies the region
Destination attributes in the outcome and their ranges	<ul style="list-style-type: none"> • Specifies the outcome
Status	<ul style="list-style-type: none"> • Has fired / has not fired
Belongs to either the source or destination context	<ul style="list-style-type: none"> • The mapping is itself an object that can be used to enrich the context
Reliability level	<ul style="list-style-type: none"> • Describes the importance or reliability of the mapping

Action Mapping Editing

This sub-module enables experts to define and edit mappings, or knowledge items, of the action type.

5 Table 5: Object range specification for source and destination attributes in action mappings

Space attribute	Source and destination spaces range specification
List variable	<ul style="list-style-type: none"> • Select more than one item in its list as compatible items • All items except those selected
Logic variable	<ul style="list-style-type: none"> • Specify 'true' or 'false' • Specify 'does not matter'

Numeric variable	<ul style="list-style-type: none"> Specify compatible range, or Specify complement of compatible range Specify 'does not matter'
Date/time variable	<ul style="list-style-type: none"> Specify compatible date or range of dates Specify complement of compatible date or range of dates Specify 'does not matter'
Text variable	<ul style="list-style-type: none"> Specify exact or partial compatible match, or Specify complement of exact or partial compatible match Specify 'does not matter'
Action object	<ul style="list-style-type: none"> Specify compatible states (specify input and/or output states or ranges) Specify complement of compatible states Specify 'does not matter' Automated activation of object or user controlled activation

The action mapping is specified by its properties as listed in Table 4.

In an action mapping, the expert specifies the objects, typically numeric and logical, that are the input to the mapping and then defines the mapping. The output is computed, not specified in a static way as for explanation mappings. The mapping is defined as follows:

- The expert selects the source space objects that go into the mapping computation.
- The expert selects or defines an operation or computation. All the system needs to support initially are the elementary mathematical and logical operations. Do Loops (Do While, Do Until, etc) may also be included as part of the 'primitive' operations. However, if one deals with low level programming, the primitive set may include the operations that are carried out on the registers in a microprocessor.
- The outcome is the result of the computation. It automatically becomes part of the destination space and part of the source space (preferably into a folder labelled as 'calculated outcomes').

The last point ensures that action mappings can be chained to produce arbitrary complex calculations. It means that the source and destination spaces overlap.

There is a similarity between the software objects mentioned earlier and the action mappings described here. An action mapping is defined in the GKMS environment whereas a software object has been defined elsewhere; it could be part of a library of procedures for example.

If an action object is part of the destination space, then the region in the source space specifies the conditions when the software object becomes part of the outcome. At access or run time an action object as part of the outcome can mean:

- Modify the object and put it in the state specified in the outcome.
- Activate the object automatically (automated activation, using the input state the object is in).
- Give user option to activate the object (user controlled activation, using the input state the object is in).

The Composite GKMS Module

A composite GKMS module is a collection of elementary GKMS modules. It comprises the same elements as the elementary GKMS module. These elements in the composite module are expressed as the union of the corresponding elements in the elementary module.

Table 6: Elements in the composite GKMS module

Element	Explanation
Source context	Union of the contexts in the elementary modules
Destination context	Union of the contexts in the elementary modules
Mappings	Union of the mappings in the elementary modules

It is frequent that the elementary modules which comprise the composite GKMS module have the same source and destination contexts.

Figure 4 illustrates the GKMS model with the source space comprising subsets of the destination space and internal state of the system. The input into the source space comprises elements of the external world, of the destination space and of the internal state. It is also possible to have the output included which becomes a subset of the destination space component

of the input. A composite GKMS module made of elementary modules with overlapping contexts is referred to a knowledge base.

A "composite GKMS" is likely to have knowledge items that have different source and destination contexts. Here we explain how the relevant information is presented to users so that they can make their own evaluation as to the validity of the context used in the knowledge item (and which affects the validity of the advice).

For users to assess the validity (if they wish) of the advice offered by the GKMS, they need to be able to see the source and destination contexts.

The source context tells users whether the person who entered the advice considered enough issues for defining the applicability of the knowledge. For example, advice about diet may use a context that does not consider the daily physical activity of the person accessing the advice. An elite sports person may decide, on inspecting the source context, that the provider of the advice did not consider all the relevant issues and discard the advice. Conversely, the sports person may decide to accept the advice, even if unexpected, if he/she can see that it the advisor did consider level of physical activity as relevant.

The destination context specifies the range of possible solutions that an advisor considers applicable to the range of problems it is dealing with. For example, an advisor may reject a range of foods (say, poultry) for animal rights reasons. A user may decide to reject the advice because, by inspecting the destination context, it can see that this advisor had a particular view about food that was affected by issues unrelated to diet (i.e. animal rights). Conversely, another user may choose to take this advice because the destination context rules out poultry food.

At the implementation level, each advice, when viewed, offers the option (via a button) to inspect the source and destination contexts associated with the knowledge item.

Concatenation Of Knowledge Bases

Composite GKMS modules can be concatenated or grouped to build larger knowledge bases. When composite modules have overlapping contexts they are described as overlapping knowledge bases. When they have non-overlapping contexts they are described as disjointed or independent knowledge bases.

When independent knowledge bases need to be concatenated, then relationships between the knowledge bases need to be made explicit. For example part of the destination context of one composite module can become part of the source space of another module. Another possibility is that both the source and destination of the two composite modules overlap.

In some situations the contexts of the knowledge bases are disjoint. In this case, the link between the two knowledge bases is done by defining another module or knowledge base which links and explains some elements of the contexts of the independent knowledge bases.

For example, see Figure 5, the destination space of one module becomes the source space, or part of the source space, of another GKMS module. Module 0 is a source GKMS to the destination modules 1 and 3 and module 2 is a source GKMS to module 0. The networking of GKMS modules gives flexibility for knowledge management and accountability. For example, module 0 could deal with the hardware aspects of a microprocessor device, module 1 with the low level software, module 2 with application software. Each module can have different domain experts who are responsible for the quality and currency of the knowledge for their module only.

Knowledge certification in networked modules can be either local or global. Local certification is when each GKMS module is certified independently of the other modules. A change to the destination space of a source module, say GKMS 0 in Figure 5, does not affect the certification of the destination module GKMS 1 and 3 in Figure 5. Global certification requires all destination modules to be re-certified when a change takes place in the destination space of a source GKMS.

At the implementation level, the expert is presented with a diagram (graphics) similar to that in Figure 5, without the connecting arrows. The expert then can add the connecting arrows to define the links between the GKMS modules. By clicking on a module, the expert is taken to the module itself and has access to all the functionality of the elementary GKMS module. For processing, the structure above is collapsed onto a single two layer structure, a single source space and a single destination space.

Query Definition Sub-Module

This sub-module enables a user to define a query in the source space, or a process to perform the role of a user. In effect, the user defines a

situation for the knowledge processing module to act on. Each object in the source space can take three values in an enquiry: i) specified, ii) 'don't know' and iii) 'unspecified'. The query definition can be one of two modes:

5 Non-interactive enquiry

 The GKMS presents the user with the complete source space. The user then specifies a situation and the GKMS looks for regions compatible with the query. If there are compatible regions, then their outcomes are presented to the user. If there isn't any compatible region, the user is informed. In this
10 case, the query can be transmitted to an interactive enquiry.

 Interactive enquiry

 The GKMS presents the user with one or a few questions at a time for the user to answer. Once it is done, the GKMS processes the answers and
15 determines the next best question(s) to ask. The best questions are those that lead to the mappings, that is regions, compatible with the query with as few questions as possible. When the system has identified a compatible region, then it presents its outcome to the user. If the system cannot find a compatible region, it informs the user.

20

 Hybrid enquiry

 The GKMS presents the user with several questions grouped logically, that is addressing a single issue. Once the user has answered these questions, the system presents the next group of questions, or single question, as the
25 case may be.

 Knowledge Processing

 Knowledge processing is the process used by GKMS to identify whether there are regions that are compatible with the enquiry. Knowledge
30 processing starts with a list of candidate regions or mappings, initially all the regions in the knowledge base, and on the basis of the objects specified in the enquiry, that is the values of the dimensions in the source space, determines:
a) the regions that are ruled out by the enquiry;
b) the regions that are compatible with the enquiry;

c) the regions that are undetermined, that is, the regions for which one cannot say whether they are compatible or ruled out because some questions have not yet been asked.

5 The process is described with reference to Figure 6. The processing stops when:

- a) there are no more candidate regions or mappings
- b) there are no more undetermined regions or mappings, or
- c) there are no more objects whose values have not been determined.

10 Processing can stop with several outcomes:

- a) There is at least one or more regions compatible with the enquiry
- b) There is no region compatible with the enquiry.

In the second case, the user is informed and the query is stored and presented to an expert who then defines a mapping, action or explanation,
15 with a region and an outcome, with the region containing the enquiry.

Location Independence

The process described above is 'location independent'. The knowledge that drives the processing resides in the mappings. This can be contrasted to
20 usual programs where the knowledge about the processing is located in the code itself and the operations depend on the location of the code in the program. Location independence is seen as a major advantage in that it frees developers from the issue of location. Processing behaviour depends only on the conditions for a mapping to take place, expressed as a region in the
25 source space. There is however an additional load put upon the system that now has to find which mapping applies next, a requirement that is taken care of in normal programming by the location of the code. Location independence is seen as particularly advantageous in 'inference chaining'.

30

Non-Interactive Processing

The GKMS takes as input the query defined by the user and looks for regions in the source space that are compatible with the query. A region is compatible with the query if the region "contains" the query. This means that
35 the value of each object in the region contains the value of the object in the query. See Table 7 below for the meaning of "contain".

This applies whether the 'does not matter' is explicit or implicit in the regions, but the system must check all objects in the source space and the dimension, number of space dimensions required to specify the region, of each region is equal to the dimension of the source space. This can be a significant computing overhead when many dimensions have values 'do not matter' in the specification of the region.

It is possible to store regions with less memory space if the dimensions that do not matter are not included explicitly in the regions. Only the dimensions with specified values are explicitly identified. At processing, unspecified dimensions of regions are taken to have the value 'does not matter'. Alternatively, and it is equivalent, the system can check that all dimensions specified in the regions are contained by the objects specified in the query. In this case a region contains a query if the query contains the compacted form of the region. This is computationally advantageous.

Table 7: Meaning of "contain"

Space attribute	Meaning of 'contain' (in A contains B)
List variable	<ul style="list-style-type: none"> List parameters in B are present in A Value of A 'does not matter'
Logic variable	<ul style="list-style-type: none"> Value of A is equal to value of B Value of A 'does not matter'
Numeric variable	<ul style="list-style-type: none"> Range of A contains the range of B Value of A 'does not matter'
Date/time variable	<ul style="list-style-type: none"> Range of A contains the range of B Value of A 'does not matter'
Text variable	<ul style="list-style-type: none"> String A contains string B, or Value of A 'does not matter'
Software object	<ul style="list-style-type: none"> Input and/or output ranges of A contain the corresponding ranges of B Input and/or output states 'do not matter'

Practically, users only define the values of the objects that are deemed relevant to the enquiry. Some objects may be left unspecified but this could mean that:

- the user does not know the value of these objects in the problem/situation at hand or
- the user erroneously considers these objects as being irrelevant to the enquiry.

Two processing approaches can be taken:

- a) 'don't knows' are treated as equal to 'unspecified' and vice versa. In this case GKMS considers all the objects or dimensions in the source space for processing. The system is able to determine which regions are compatible with the enquiry. All the other regions are deemed incompatible with the enquiry.
- b) 'don't knows' are different from 'unspecified'. In this case the GKMS only considers the objects whose values have been specified for processing. On the basis of this limited number of dimensions, the system (GKMS) determines which regions are compatible, which are ruled out and which are still candidates:
 - ruled out regions are those that are not compatible with the enquiry on the basis of the specified objects in the enquiry.
 - compatible regions are those that are compatible with the enquiry on the basis of the specified objects in the enquiry and for which all the unspecified objects or dimensions have values (in the region) equal to 'does not matter'.
 - candidate regions are those that are compatible with the enquiry on the basis of the specified objects in the enquiry and for which some of the unspecified objects or dimensions have values (in the region) that are not 'does not matter'.

When some candidate regions still exist, the system moves to an interactive processing mode to determine whether some of these candidate regions may be compatible with a more precise enquiry, as will now be described.

Interactive Processing

The system calculates the 'discriminating power' of each object, or dimension, in the source space the value of which has not yet been specified. The discriminating power can be calculated in 2 ways:

- a) Calculate the number of regions, or mappings, for which the value of the object does matter. The discriminating power is the number divided by the number of regions in the candidate list. This assumes that the number of regions for each of the legal values of the object is approximately the same.
- b) Calculate the number of regions, or mappings, for which the value of the object does matter, for each possible value of the object; the range of

values for each object is divided in segments. The discriminating power is the average number of regions per segment, multiplied by the number of segments and divided by the number of regions in the candidate list. This number is weighted with (in the first instance, divided by) its variance (the lower the variance the higher the discriminating power).

Table 8: Segment identification in the calculation of the discriminating power

Space attribute	Segment identification
List variable	<ul style="list-style-type: none"> Each list parameter (item on the list) is a segment Number of segments = number of list parameters
Logic variable	<ul style="list-style-type: none"> The segments are: 'True', 'False' and 'Does not matter' 3 segments
Numeric variable	<ul style="list-style-type: none"> The legal range of the variable is divided in non-overlapping segments The segments boundaries are defined by the end-points of the ranges for each of the regions (for which the variable matters) A region range can be made of several segments A segment can belong to several regions
Date/time variable	<ul style="list-style-type: none"> As for numeric variables
Text variable	<ul style="list-style-type: none"> The segments are: 'Contains', 'Does not contain' and 'Does not matter' Alternatively, a text variable can be treated as a list variable, with the list parameters being the string characters
Software object	<ul style="list-style-type: none"> As for numeric variables (when 1 variable only) Complicated when there is more than 1 variable

Once the discriminating power of each object is calculated, the system asks the question with the highest discriminating power. If several objects have the same discriminating power, then the system either presents all the related questions or selects one of them only for presentation to the user. Once the user has supplied the answer related to the object with the highest discriminating power, the system uses the answer to remove from the list of candidate all the knowledge items that are ruled out by the answer. The

process described above is then repeated, that is, the discriminating power of each non-specified object or dimensions in the remaining candidates is then calculated, etc. This is carried out until there are no more candidate regions, no more undetermined regions or no more dimensions for which the values have not been ascertained.

Forward And Backward Chaining

Interactive processing can be either forward chaining or backward chaining.

Forward chaining processing selects, via the interactive question/answer session, the next best question that will identify the region, if any, that contains the query.

Backward chaining processing selects, via the interactive question/answer session, the next best question that will identify the region, if any, that is compatible with the desired outcome. With backward chaining, users can define a desired outcome by selecting from the objects in the destination space and then find out whether their situation applies. The desired outcome may require the situation to be described by more than one region. In backward chaining the system calculates the discriminating power of source dimensions as follows:

a) Calculate the number of knowledge items for which the value of the objects in the desired outcome do matter. The discriminating power is the number divided by the total number of outcomes. This assumes that the number of outcomes for each of the legal values of the object is approximately the same.

b) Calculate the number of knowledge items for each possible value of the objects in the desired outcome; each object value can be divided in segments. The discriminating power is the average number of regions per segment weighted with (in the first instance, divided by) its variance (the lower the variance the higher the discriminating power).

Non-interactive processing can also be backward. In this case users define a desired outcome and the systems informs them of the regions, if any, that are compatible with the stated outcome. The system does that by finding which existing outcomes in the knowledge base contain the target outcome and then selects the ones with the number of stated dimensions nearest to the number of stated dimensions in the target outcome. If no region is

compatible, the system can calculate which dimensions of the outcome would need to be changed for the modified outcome to be compatible with a region, by the system finding which outcomes in the knowledge base has the nearest number of compatible dimensions, either too many or too few, with
5 the desired outcome. The system then presents these outcomes.

Hybrid Processing

Hybrid processing takes place when a user starts by defining an enquiry in a non-interactive way, using a subset of all the dimensions in the
10 source space, either the system presents only a subset or the user makes use of only a subset. The system then selects the knowledge items compatible with the enquiry and, if there are more than one, guides the user to the most appropriate knowledge item, if any, using interactive processing.

15 Sub-Regions

It can happen that the system identifies one or several regions that are compatible with the enquiry. The system may also detect that there are still regions that are candidates with the enquiry, that is, knowledge items that have not been eliminated on the basis of the information supplied by the
20 user, either by specification of an enquiry in a non-interactive way or by answering the questions asked by the system so far. In this case, the system moves into a non-interactive processing mode to determine which is the best next question to ask in order to identify the most appropriate knowledge item. This situation happens when not all dimensions have been specified,
25 and also when a region has sub-regions. While the region is compatible with the enquiry defined so far, a more precise answer may be given if the problem situation can be determined to, in fact, belong to one of the sub-regions. The system identifies sub-regions by checking whether a region contains another (see Table 4 for the meaning of 'contain').

30

Inference Chaining

Inference chaining uses inferred objects, inferred objects are a members of the destination space, attached to primary regions $r(i)$ in the source space, which are themselves part of inferred regions, say $r(j)$, in the
35 source space. When the value of an inferred object is 'true' or valid, then the inferred region $r(j)$ it is part of in the source space can become part of an

answer to an enquiry if the other required conditions are satisfied. The outcome attached to inferred region $r(j)$ is then the (or a possible) answer to the enquiry. This chaining effect can be repeated as many times as an expert wishes.

- 5 Inference chaining applies to explanation and action mappings.
As explained above, the property of 'location independence' of GKMS processing is a major advantage for software developers in that it frees them from the need to consider not only the code but also its location to determine its impact on the behaviour of the system.

10

Collapsing Regions

- Inference chaining is based on objects whose value is the outcome of an inference. A question arises regarding their discriminating power. The discriminating power of inferred objects does not need to be calculated. Only
15 the discriminating power of non-inferred objects is calculated. An outcome attached to a region which has inferred objects is selected when its 'collapsed equivalent' region is compatible with the enquiry. The 'collapsed equivalent' region is the region obtained in the source space when all the inferred objects are replaced by the objects in the region to which they, the inferred objects,
20 belong. By collapsing the regions, one can also determine whether the inference chaining is consistent. That is, whether some objects in the inferred regions are required to take two different values for the chain to be inferable to the end. That is whether a primary region requires a certain value to an inferred object to be activated which in is a member of another
25 region that requires the same object to take a different, incompatible value for the inference chain to continue.

Alternative Knowledge Processing

Knowledge processing can be carried out in a few steps:

- 30 1. Identify the ruled-out regions (that is, identify the compatible and undetermined regions).
2. Discriminate between compatible and undetermined regions.
3. Order compatible regions according to their specificity to the questions asked and answered so far.

4. Order the undetermined regions according to their relevance to the enquiry defined so far.
5. Select and order the questions that should be asked next.

- 5 1. Identify the compatible and undetermined regions (that is, the regions which are not ruled-out). This is done in three steps, for each knowledge item in the candidate list:

10 Step 1: Identify the regions (or knowledge items) which **can** be compatible or undetermined regions. This is done by identifying the regions in which at least one object (question) is present in the source region of the knowledge item with the correct value (that is, with the value given in the answer being part of the region).

15 Criterion 1: object1-value1 |
object2-value2 | ... | objectN-valueN

This reads as: For a region
(knowledge item) to be compatible or undetermined, one
needs:

20 object1 with value1 in enquiry is present in candidate
region, or

object2 with
value2 in enquiry is present in candidate region, or

25 ...
objectN with

valueN in enquiry is present in candidate region

30 For all the objects specified as part of the enquiry (in our example: N)

The result of step 1 is a collection of knowledge items (collection1).

35 Step 2: Identify the regions (or knowledge items) in the candidate list that can be ruled-out. This is done by identifying the regions in which one object (question) is present in the source region of the knowledge

item but with the wrong value (that is, the value given in the answer to the question in not part of the region).

Criterion 2: (object1 ¬
 5 object1-value1) | (object2 ¬ object2-value2) | ...
 ... | (objectN ¬ objectN-
 valueN)

This read as: For a region
 10 (knowledge item) to be ruled-out, one needs:

object1 is present in the candidate region but with the
 wrong value, or
 object2 is present
 15 in the candidate region but with the wrong value, or
 ...
 objectN is
 present in the candidate region but with the wrong value

20 For all the objects specified as part of the enquiry (in our example: N)

The result of step 2 is a collection of knowledge items (collection2).

Step 3: The collection of knowledge items that need to be kept as
 25 compatible and undetermined is given by:

Criterion 3: (Criterion 1) &
 not (Criterion 2)

30 collection = knowledge items present in collection1 but not present in
 collection2

Implementation:

35 In the Lotus Notes/Domino environment, the above can be implemented
 using the FTSearch procedure.

Discriminate between compatible and undetermined regions (or knowledge items)

5

Compatible and undetermined regions are discriminated by determining the score of each region (or knowledge item) present in the collection returned by 1 above.

10

The score of a region is given by the number of questions or defined objects in the enquiry that are present in the region under consideration. From that the compatible and undetermined regions can be identified by comparing the score with the number of defined objects in the knowledge item regions.

15

Compatible regions: Score = number
of defined objects in the knowledge item region

Candidate regions: Score < number
of defined objects in the knowledge item region

20

The score of each region is calculated as follows:

$$\text{Score} = \text{object1-value1} + \text{object2-value2} + \dots + \text{objectN-valueN}$$

25

The “+” between the objectX-valueX groups is the operator “accrue”. It adds the number of time a group is present in the region. In some cases the adding step is weighted with each group having a different weight.

Note that it would be possible to replace step1 in section 1 above by:

30

1. calculating the score of each knowledge item as shown in this section
2. keeping only the knowledge items with Score > 0.

35

The best method depends on the way the computation is carried out and what tools are available in the computing environment. Can all the knowledge items be processed in one step (this can be quite efficient), or

does one need to take each knowledge item in the knowledge base one after the other (this can be quite slow).

Implementation:

In the Lotus Notes/Domino environment, the above can be implemented using the FTSearch procedure.

3. Order compatible regions according to their specificity

The specificity of a compatible knowledge item (or region) with respect to an enquiry is defined as the number of objects in the region divided by the number of attributes in the enquiry.

$$\text{Specificity} = (\text{n objects in region}) / (\text{n objects in enquiry})$$

4. Order the undetermined regions according to their relevance

Undetermined knowledge items (or regions) can be calculated in two ways.

- a. The higher the proportion of objects in the region already accounted for by the questions answered, the higher the relevance of the region.

$$\text{Relevance} = (\text{n objects in regions accounted for}) / (\text{n objects in region})$$

- b. The fewer the number of questions left to answer to establish whether a region is compatible or ruled-out, the higher the relevance.

$$\text{Relevance} = (\text{n objects in region}) - (\text{n objects in enquiry that are present in region})$$

In some cases this would not work well. For example, a region could have the same relevance as another but with a larger number of objects in its region. One could argue that it is proportionally closer to being fully answered than the first one.

5

5. Order the supplementary questions

When there are undetermined knowledge items (or regions) then their status (either compatible or ruled-out) can be determined by asking further questions. These questions can be ordered according to the number of time in which the objects they relate to (a question relate to an object and is asked so that a value can be given to this object for an enquiry) appear in the regions that belong to the undetermined list. Note that it is not necessary to consider the objects (questions) that have already been answered.

6. Multi-choice selection lists in enquiries

Sections 1 and 2 above deal with questions which allow users to define (usually select) a single value or range. There are situations where users may wish to define or select more than one value. This section deals with the changes that are required to knowledge processing to deal with such situations.

	<u>Section 1, step 1:</u> be modified	does not need to
	<u>Section 1, step 2:</u> modified	needs to be
25	<u>Section 3:</u> be modified	does not need to
	<u>Section 4:</u> be modified	does not need to
30	<u>Section 5:</u> be modified	does not need to

Modification to Section 1, step 2:

At present step 2 identifies the regions (or knowledge items) in the candidate list that can be ruled-out. This is done by identifying the regions in which one object (question) is present in the source region

35

of the knowledge item but with the wrong value (that is, the value given in the answer to the question is not part of the region).

To take account of the multiple values, this needs to be changed to: identify the regions in which one object (question) is present in the source region of the knowledge item but with not even one of the values selected is part of the region.

```

(object1 &not ( object1-value1,1 | object1-value1,2 | ... | object1-
value1,M )) |
(object2 &not ( object2-value2,1 | object2-value2,2 | ... | object2-
value2,M )) |
...
(objectN &not ( objectN-valueN,1 | objectN-valueN,2 | ... |
objectN-valueN,M ))

```

This read as: For a region
(knowledge item) to be ruled-out, one needs:

object1 is present in the candidate region and not one of the values selected in the enquiry is present in the candidate region, or

object2 is present in the candidate region and not one of the values selected in the enquiry is present in the candidate region, or ...

objectN is present in the candidate region and not one of the values selected in the enquiry is present in the candidate region

For all the objects specified as part of the enquiry (in our example: N)

There are also other ways to implement knowledge processing.

Access To All The Knowledge

The knowledge base is made of knowledge items, each comprising a set of objects that are used to define their regions in the source context. The issue is to ensure that at least one question is asked about each knowledge item, so that the GKMS, using the answer given by the user, can determine whether the region (that is, the knowledge item) is ruled-out, undetermined or compatible.

(If the region has a single object, then one question suffices to determine whether the knowledge item is ruled-out or compatible. If the region has more than one object, then, on the basis of one question, the GKMS can only determine whether the knowledge item is ruled-out or is undetermined.

5 However, by asking further questions, the final status of the knowledge item can be ascertained.)

Procedure for questions selection:

The best first question to ask (that is, the objects in the source context to select as first question) can be determined by calculating the number of
10 knowledge items each object in the source context belongs to, and to order these objects in descending order. The object that appears in the largest number of regions is the best question to ask. Asking this question accounts for all the knowledge items the object corresponding to the question belongs to.

15 The second best question to ask can be determined as follows: take the set of knowledge items not accounted for by the first question and carry out the process described in the previous paragraph.

The next best questions can be determined by repeating this process, until all the knowledge items in the database have been accounted for.

20 It may happen that there are too many independent questions to account for the knowledge base (that is, to reach all the knowledge items in the knowledge base). In this case one can define a new object in the source context that can become a member of a number of (perhaps all) knowledge items (that is, it belongs to their regions), with the knowledge items being
25 grouped according to the values that this object can take. This one object now accounts for a large number of knowledge items.

For example, in a knowledge base for trouble-shooting a car, there may be many unrelated questions about the electrical aspects of the car. In this case, one can introduce a new question (object) that enables one to determine
30 which part of the electrical system is functioning or not (before going into the details of why it is not functioning). By making this new object a member of the regions dealing with electrical issues, one effectively reaches a large number of previously independent knowledge items by asking one question. (That is, one question accounts for a large number of knowledge items that
35 could only be reached previously by asking many questions.)

Questions presentation:

The first few best questions are presented on the first screen. If there too many questions to fit on one screen, then the next best questions can be asked on the second or third screens, in addition to any supplementary questions raised by the answers to the questions asked so far.

5

System Behaviour

The system behaviour is specified by considering the state the system is in and in taking actions based on this state. The state of the system can be included as part of the source space and the outcomes that manipulate the state of the system can be part of the destination space. Alternatively, the system behaviour can be specified using a separate module that has as source space a subset of the source space mentioned above and as destination space a subset of the destination space mentioned above.

The system behaviour module enables the expert to define the behaviour of the system it is attached to. It offers an alternative way to implement interactive processing, where the system decides which questions to ask next, in which the expert specifies what the system should do under defined conditions. The system behaviour module is optional.

The system behaviour module is in an elementary GKMS module which consists of the same source space as that of the system it is attached to and a different destination space. The state of the system is described as in the elementary GKMS module, using regions attached to outcomes of specified behaviours, the destination space. The destination space specifies behaviour options for the system and has the same attributes or objects as the source space but with different possible values for actions specification, and some other actions as well. Table 9 shows the destination space of a behaviour module.

Table 9: System behaviour destination space

Space attribute	Destination spaces value options
List variable	<ul style="list-style-type: none"> • Display variable as enquiry • Hide variable
Logic variable	<ul style="list-style-type: none"> • Display variable as enquiry
Numeric variable	<ul style="list-style-type: none"> • Display variable as enquiry • Hide variable • Specify legal range

Date/time variable	<ul style="list-style-type: none"> · Display variable as enquiry · Hide variable
Text variable	<ul style="list-style-type: none"> · Display variable as enquiry · Hide variable
Software object	<ul style="list-style-type: none"> · Display variable as enquiry · Hide variable · Specify state (specify input and/or output states or ranges)
Other	<ul style="list-style-type: none"> · Save enquiry · Activate processing (re: search for advice) · Hide all variables displayed after variable x (variables in the display can also be identified with a number) · Display another object which is not part of the problem space (text, image, video, etc) as advice of supplementary information · Abort process

Knowledge Review

Experts need to be able to find out conveniently what knowledge items mention some objects. To find that out the expert defines a query which specifies objects and their values, in the source and destination spaces and instructs the system to retrieve all knowledge items that are contained by the enquiry; the unspecified objects are assumed to have the value 'does not matter'. The expert can then review, update and deactivate these knowledge items.

Overlapping Knowledge Items

As shown in Figure 1, knowledge items can overlap and when this occurs experts need to know. Overlap can be in the source space or in the destination space. Overlap happens when a region contains part of another region. Sub-region is a special case of overlapping regions in which one region contains the whole of another region. The meaning of 'overlap' is explained in Table 10 (see Table 7 - meaning of 'contain' - for comparison).

Table 10: Meaning of 'overlap'

Space attribute	Meaning of 'overlap' (in A overlap with B)
-----------------	--

List variable	<ul style="list-style-type: none"> Some list parameters in B are present in A Value of A or B 'does not matter'
Logic variable	<ul style="list-style-type: none"> Value of A is equal to value of b Value of A or B 'does not matter'
Numeric variable	<ul style="list-style-type: none"> Range of A overlaps with the range of B Value of A or B 'does not matter'
Date/time variable	<ul style="list-style-type: none"> Range of A overlaps with the range of B Value of A or B 'does not matter'
Text variable	<ul style="list-style-type: none"> String A overlaps with string B (some characters are common) Value of A or B 'does not matter'
Software object	<ul style="list-style-type: none"> Input and/or output ranges of A overlaps with the corresponding ranges of B Input and/or output states 'do not matter'

Overlap can be checked at any time by the user but can also be checked automatically whenever a new knowledge is defined or an existing knowledge item is modified, that is, whenever the knowledge base is modified.

Knowledge Certification

With reference to Figure 1, knowledge items are certified with respect to explicit source and destination contexts. When the context is modified, knowledge items can still be used as long as their context is made explicit to the users. Alternatively, the knowledge items may need to be re-certified by experts. The expert may need to modify some knowledge items before re-certification.

The GKMS Module Architecture

The architecture is shown in Figure 7 which lists all the elements mentioned.

Practical Implementation Issues

In the section below the emphasis is on solutions which support the GKMS model and which are both computationally effective as well as user

friendly. In all cases, the implementation is designed to enable non-computer specialists to build and use knowledge systems.

Context Editing

5 Context editing needs to be done in two situations:

1. To define an initial domain of discourse before any knowledge items have been defined
2. To modify, usually extend, a domain of discourse to enable it to deal with an extended range of situations and problems.

10

Initial Context Editing

Users are presented with a blank context, or a context with one or two objects in it as examples. Users can then:

- Define an object or attribute
- 15 • Edit an existing attribute
- Delete an existing attribute

Modification Of An Existing Context

20 An existing context can be edited as explained explained above. A second way of editing the context is in response to new queries which require its modification, usually extension. This second way is treated below as it is part of knowledge acquisition in a variable context.

Knowledge Acquisition

25 Knowledge acquisition is also referred to as mappings definition.

Acquisition In A Fixed Context

The context has been defined by domain experts to their satisfaction. That is, the domain of discourse has been set and the task is now to enter
30 knowledge in the form of mappings.

Acquisition Not Prompted By Enquiries

In this situation the experts use their experience to decide, without external prompts, what mappings are needed. They then define these
35 mappings by defining regions in the source context and outcomes in the destination context. The mapping definition process is to define a situation

or problem in the source context and then to specify the outcome corresponding to the problem. An alternative way is to define a solution which is known to be useful or relevant to the domain of discourse and then to specify the regions which determine when this outcome is applicable. In practice the two approaches can often be merged.

Knowledge acquisition is a variant of the process described below. The process is as follows:

1. Define a region in the source context
2. Define an outcome in the destination context
3. Save knowledge item (specify a title and a summary)

Steps 1 and 2 can be interchanged. At any stage in the process it is possible to go from any step to any other step.

Acquisition Prompted By Enquiries

In this situation, an enquiry, typically by a user who is not a domain expert, either has no solution or a solution which is deemed to be inadequate. A enquiry which has no solution is an enquiry for which there is no compatible region in the GKMS. This enquiry is then routed to a domain expert which uses it as a prompt for adding new knowledge, in the form of new mappings, to the system. This process is illustrated in Figure 2, Part 2. The process in Figure 2 can be guided by the GKMS as follows:

1. GKMS presents a list of unanswered queries to a domain expert
 - The unanswered enquiries are presented as a view, with date, and reason for being unanswered .
2. The expert selects an enquiry and opens it.
 - The enquiry is presented in the source context form (SCon1)
 - The instruction on Scon1 is: "please inspect the enquiry and when ready press "next" to answer it". Scon1 cannot be edited.
3. The domain expert answers the query
 - GKMS presents the destination context (Dcon1). Depending on the size of the display screen the two forms Scon1 and Dcon1 can be shown simultaneously. Dcon1 is editable.
 - The instruction on Dcon1 is: "please enter the solution to the enquiry by specifying the relevant attributes and providing explanations for your answers.

- The expert defines the solution to the enquiry.
 - The expert can proceed by pressing “next” or can go back to the enquiry (Scon1) by pressing “back”. The only other option available to the expert is to abort the mapping definition process (by pressing the “cancel” button).
 - 5 • Pressing “next” takes the expert to the generalisation process (see below)
 - 4. Return to source context to generalise the enquiry to a region if possible
 - GKMS presents Scon2. Scon2 is identical to Scon1 except that it can be edited.
 - The instruction on Scon2 is: “Generalise the enquiry if possible by
 - 10 defining a region “around” the enquiry for which the answer specified on the previous screen applies”.
 - The expert can: a) go back to the Dcon1 to inspect or edit the solution just defined, b) press “next” to continue, or c) press “cancel” to abort the mapping process.
 - 15 • Pressing “next” allows the knowledge item (or mapping) to be saved.
 - 5. Save knowledge item
 - GKMS presents the mapping form (Mform) which enables the expert to enter the title for the mapping and a summary. Only the title is mandatory. When finished the option is to press “next” or “back” or “cancel”.
 - 20 • With “back” the expert can go back to the previous display. “Cancel” aborts the knowledge mapping process.
 - With “next” GKMS displays: “this knowledge item will be added to the knowledge base and will be used to answer future enquiries. OK, back or cancel”.
 - 25 • “OK” save the mapping, “back” goes back to the previous screen, “cancel” aborts the knowledge definition process.
- When an enquiry has been answered in the way described above, it is taken off the list of unanswered queries and the view (see point 1 above) is updated.
- 30 At any time during steps described above, the expert can return to either the source context with the problem in it (no editing allowed), the editable source context with the region in it (to modify the region), or to the destination context to modify the solution proposed.
- The process described above can be varied by exchanging steps 3 and 4, as
- 35 shown below:
1. GKMS presents a list of unanswered queries to a domain expert

2. The domain expert inspects the enquiry
3. The domain expert generalises the enquiry to a region if possible
4. The domain expert answers the query
5. Save knowledge item

In a practical implementation the forms can all be presented to the expert, with only the ones that are intended to be edited in the editing mode. The advantage of this is that the expert can always have a synoptic view of the knowledge item being defined.

Acquisition In A Variable Context

The section below recognises that mappings may have different source and destination contexts. The acquisition process for variable contexts builds on the acquisition process in a fixed context described above. The extra or modified steps are shown below in *italic*.

1. GKMS presents a list of unanswered queries to a domain expert
 - The unanswered enquiries are presented as a view, with date, and reason for being unanswered .
 - The expert selects an enquiry and opens it.
2. The domain expert inspect the enquiry
 - The enquiry is presented in the source context form (SCon1)
 - The instruction on Scon1 is: "please inspect the enquiry and when ready press "next" to answer it". Scon1 cannot be edited.
 - The instruction on Scon1 is: "please inspect the enquiry and when ready press "next" to answer it". Scon1 cannot be edited.
 - *The enquiry may be accompanied by a message from the user who filed the enquiry. The message may describe the enquiry more specifically or in more details (see "unanswered or poorly answered enquiries").*
 - *Based on this message or on professional judgement, the expert may decide to extend (or reduce) the source context. This is described in point 4b. below.*
 - *It is also possible to allow source context editing at this stage in the process:*
 - *Scon1 has an additional button "edit context"*
 - *pressing "edit context" allows Scon1 to be edited (the enquiry cannot be modified, the attributes in the enquiry cannot be deselected -the deselect option is added to Scon1, new attributes can be added.*
 - *see point 4b below for details.*
 - *Scon1, when shown at this stage in the process, has the message: "please*

modify context as appropriate without changing the enquiry. Press "next" when ready to answer it".

- *when finished editing the context the user can press "next" to move to the*

5 *destination context to answer the enquiry.*

3. The domain expert answers the query

- GKMS presents the destination context (Dcon1). Depending on the size of the display screen the two forms Scon1 and Dcon1 can be shown simultaneously. Dcon1 is editable.

10 • The instruction on Dcon1 is: "please enter the solution to the enquiry by specifying the relevant attributes and providing explanations for your answers.

- The expert defines the solution to the enquiry.

15 • The expert can proceed by pressing "next" or can go back to the enquiry (Scon1) by pressing "back". The only other option available to the expert is to abort the mapping definition process (by pressing the "cancel" button).

3b. Specify appropriate destination context

This involves either reducing the context or enlarging the context. Reducing the destination context corresponds to considering fewer options for the outcome than there is present in Dcon1. Enlarging the context corresponds to adding

20 *new outcome options to Dcon1.*

Dcon1 has one additional button labelled "add new attribute".

Reducing the context

- *By default all the attributes in Dcon1 are part of the mapping's context.*

25 • *The expert can deselect some of these attributes by "unchecking" it (each attribute can have a check box for selection purposes for example).*

Enlarging the context

- *Pressing the "add new attribute" button opens the context editing module (see section above).*

30 • *The destination context the expert can add to is the Dcon1.*

- *Pressing "next" takes the expert to the generalisation process (see below)*

4. Return to source context to generalise the enquiry to a region if possible

- GKMS presents Scon2. Scon2 is identical to Scon1 except that it can be edited.

- The instruction on Scon2 is: "Generalise the enquiry if possible by defining a region "around" the enquiry for which the answer specified on the previous screen applies".

4b. Specify appropriate source context

- 5 *Reducing the source context Scon2 corresponds to considering fewer attributes for defining the enquiry and accepting fewer attribute which are not specified in the enquiry (that is, which do not play a role in the definition of the enquiry). Enlarging Scon2 corresponds to adding new attributes for enquiry definition and for specifying which of these attributes do not play a role in the*
- 10 *enquiry.*

Scon2 has one additional button labelled "add new attribute".

Reducing the context

- *By default all the attributes in Scon2 are part of the mapping's context.*
 - *The expert can deselect some of these attributes by "unchecking" it (each*
- 15 *attribute can have a check box for selection purposes for example).*

Enlarging the context

- *Pressing the "add new attribute" button opens the context editing module (see section above).*
- *The destination context the expert can add to is Scon2.*

20 *4c. Review the region*

- *The process is the same as in point 4. Above.*
- *The expert can: a) go back to the Dcon1 to inspect or edit the solution just defined, b) press "next" to continue, or c) press "cancel" to abort the mapping process.*

- 25 • *Pressing "next" allows the knowledge item (or mapping) to be saved.*

5. Save knowledge item

- *GKMS presents the mapping form (Mform) which enables the expert to enter the title for the mapping and a summary. Only the title is mandatory. When finished the option is to press "next" or "back" or "cancel".*

- 30 • *With "back" the expert can go back to the previous display. "Cancel" aborts the knowledge mapping process.*

- *With "next" GKMS displays: "this knowledge item will be added to the knowledge base and will be used to answer future enquiries. OK, back or cancel".*

- 35 • *"OK" save the mapping, "back" goes back to the previous screen, "cancel" aborts the knowledge definition process.*

- *Each knowledge item is saved with its source and destination context.*

At any time during steps described above, the expert can return to either the source context with the problem in it (no editing allowed), the editable source context with the region in it, to modify the region, or to the destination context to modify the solution proposed. In a practical implementation the forms can all be presented to the expert, with only the ones that are intended to be edited in the editing mode.

Knowledge Access

Interactive and non-interactive knowledge access have already been described. Here we consider the action which activates a knowledge search in an GKMS and the process used to presenting the outcomes of the knowledge items or mappings which are compatible with the enquiry.

Triggering Mechanisms

The access process can be triggered by a user who defines an enquiry or by a system enquiry, that is, an enquiry specified by values produced by another package or process in a computer system.

Access Mechanisms

In a similar way to the triggering mechanisms, access can be by presentation to a user on a screen, for explanation mappings, or by running some processes or modules, for action mappings. The way the final results are produced and presented to users is under the control of the processes or modules in the outcomes.

Ranking Of Mappings And Their Outcomes

Knowledge items or mappings need to be ordered according to their fit with the enquiries. This applies to definite outcomes only or to definite and candidates outcomes; that is to definite and candidate mappings. Rejected mappings are not considered here. The fit is determined by three factors:

1. Weight of each source attribute, weights may not all be equal.
2. For definite and candidate knowledge items: proportion of attributes in the enquiry which are satisfied by the region attributes; that is attributes belonging to the region of the knowledge item.

3. For definite and candidate knowledge items: proportion of the attributes in the source context which belong to the region of a candidate knowledge item.

5 Weight Of Source Attributes

a(i) attribute i

w(i) weight of attribute i

nw(i) normalised weight of attribute i

Sum w(i) over all attributes = Sw(i)

10 nw(i) = w(i) / Sw(i) normalisation is not strictly necessary.

Proportion Of Enquiry Attributes Satisfied By Region Attributes

This specifies how general or specific a knowledge item is in
15 addressing an enquiry. A high proportion $1/p_1$ indicates a general knowledge item applicable to many enquiries. A low proportion indicates a well targeted answer. In practice is easier to use the inverse p_1 of the proportion $1/p_1$ ($p_1 \leq 1$). A value of p_1 close to one indicates specific and precise knowledge and is highly desirable; a lower value is less desirable.
20 This measure does not consider whether a knowledge item is a candidate or a definite, which is addressed below.

$$1/p_1 = Na(i,q) / (Na(i,r,q)) ; 1/p_1 \geq 1$$

Na(i,q) = number of attributes i in enquiry q

25 Na(i,r,q) = number of attributes in the region r of a knowledge item which belong to the enquiry q

With attribute weights:

$$1/p_1 = Nw(i,q) / (Nw(i,r,q)) ; 1/p_1 \geq 1$$

Figure 8 illustrates the situations for $p_1 \leq 1$; attributes in the region that "do not matter" are not specified. A simple implementation of this
30 ranking is the number of attributes in the region. Knowledge items which are found to be compatible with the enquiry and which have larger numbers of attributes are more specific than knowledge items with few attributes in their regions.

35 Proportion Of A Mapping's Source Attributes Which Satisfy The Enquiry

This measure applies to candidate mappings. It determines how close candidate mappings are to becoming either rejected mappings or definite mappings. For definite mappings this proportion is 1. The inverse of the desired proportion p_2 is first calculated. $p_2 \leq 1$.

5 $1/p_2 = Na(i,r) / (Na(i,r,q)) ; \quad 1/p_2 \geq 1$
 $Na(i,s) =$ number of attributes i in the region r of a knowledge item
 $Na(i,r,q) =$ number of attributes in the region r of a knowledge item
 which belong to the enquiry q

With attribute weights:

10 $1/p_2 = Nw(i,r) / (Nw(i,r,q)) ; \quad 1/p_2 \geq 1$
 $p_2 = 1$ for definite knowledge items
 $p_2 < 1$ for candidate knowledge items

Figure 9 illustrates the situation for $p_2 \leq 1$, and attributes that "do not matter" have not been specified.

15

Overall Ranking

The relevance of a knowledge item, which determines its order or ranking among the knowledge items compatible with an enquiry, candidate of definite knowledge items, is calculated as:

20 For definite knowledge items: $R = p_1$
 For candidate knowledge items: $R = p_1 * p_2$

A relevance ranking can also include the probability or reliability level p_3 that the outcome of a knowledge item solves the situations that can be described in its region. In this case we have:

25 For definite knowledge items: $R = p_1 * p_3$
 For candidate knowledge items: $R = p_1 * p_2 * p_3$

Related Issues

30 Fuzzy logic principles can also be included in GKMS. For example, a query can cover several overlapping regions of several knowledge items. Similarly, outcomes can overlap. The impact on relevance can be calculated taking into account the overlaps and the weights of the attributes and similar considerations above.

35 Unanswered Or Poorly Answered Enquiries

An unanswered enquiry has no compatible mappings and therefore no outcomes. A poorly answered enquiry can be detected in several ways, which can be specified by users or domain experts:

- The definite knowledge items (and therefore outcomes) have a poor relevance ranking.
- There are candidate knowledge items but no definite knowledge items.
- The user deems the outcome as unsatisfactory and sends feedback to the GKMS administrator.

Once detected, GKMS deals with unanswered or poorly answered enquiries by referring them to an expert who can then add new knowledge in the GKMS to: a) answer this specific enquiry, and b) answer any future enquiry which falls within the region of the newly created knowledge item.

Knowledge Management For Domain Experts

Domain experts need tools to enable them to manage the knowledge items or mappings which have been entered into a GKMS. This becomes necessary as soon as more than a few mappings have been entered.

Domain experts need to manage the knowledge base from the point of view of knowledge comprehensiveness, consistency, quality, etc. The tools available to assist them are:

- Detect overlapping knowledge items, in their source and/or destination
- Identify knowledge items with a specified combination of source and destination attributes or values
- Edit existing knowledge items
- Inspect history of knowledge items in the system
 - list of knowledge items (active and deactivated)
 - when created, by whom

Overlapping Knowledge Items

Given a certain knowledge item, this tool detects which other knowledge items or mappings in the GKMS overlap with it. The overlap can be in the regions of the mappings or in their outcomes. This tool enables the domain expert to check that overlapping knowledge items are compatible.

With overlapping regions, an enquiry in the overlapping part or the regions will produce more than one outcome (one outcome per overlapping

region). These outcomes need to be compatible (that is, not contradictory) for the system to provide meaningful results.

With overlapping outcomes, the domain expert can determine which conditions (i.e. regions) produce these outcomes and determine whether these different conditions are not mutually exclusive or contradictory for example.

If overlapping knowledge items are found to be incompatible in some of the ways described above, then the domain experts need to edit one or several of them.

Overlap Detection In Fixed Common Contexts

Here we consider all knowledge items or mappings to have the same source and destination contexts. Note that we assume that the regions are defined by attributes which have specified values. The attributes which do not matter are not included in the region definitions.

Region overlap detection

The process for region overlap detection is as follows:

1. The expert selects which knowledge item to use as reference for detecting overlaps.
2. When the expert click the button "detect overlap", the GKMS:
 - Takes the region of the reference knowledge item.
 - Transforms this region into an enquiry (that is, GKMS displays it in the enquiry specification environment).
 - Activates the search for knowledge item (the same search as when searching for outcomes compatible with an enquiry; it searches for knowledge items with regions which are compatible with the enquiry). The reference knowledge item is not included in the list of knowledge items searched.
3. The GKMS search engine retrieves and presents the knowledge items with outcomes which are definite or candidates. This is different from a normal search in which the system only presents the outcomes).
4. The expert can then inspect and edit these knowledge items.

Outcome overlap detection

The process for outcome overlap detection is as follows:

1. The expert selects which knowledge item to use as reference for detecting overlaps.

2. When the expert click the button “detect overlap”, the GKMS:
 - Takes the outcome of the reference knowledge item.
 - Transforms this outcome (or region in the destination space) into an enquiry (that is, GKMS displays it in the enquiry specification environment).
 - 5 - Activates the search for knowledge item (the same search as when searching for outcomes compatible with an enquiry; it searches for knowledge items with outcomes (not regions in the source space as in a normal search) which are compatible with the enquiry). The reference knowledge item is not included in the list of knowledge items searched.
 - 10 3. The GKMS search engine retrieves and presents the knowledge items which are definite or candidates. This is different from a normal search in which the system only presents the outcomes).
 4. The expert can then inspect and edit these knowledge items.
- In practice, experts often search for knowledge items which overlap in their
- 15 sources and destinations.

Overlap Detection In Variable Contexts

Here we consider all knowledge items or mappings which can have different source and destination contexts.

20

Search For Knowledge Items With Specific Characteristics

Experts can search the knowledge base of the GKMS for knowledge items which meet specified characteristics. The process is as follows:

1. The GKMS presents the expert with the source and destination contexts.
- 25 2. The expert defines an query in these contexts by specifying values for attributes in one or both of the contexts.
3. Pressing “detect overlap” activates the search for overlapping items. The search algorithm combines the two searches described above, that is the GKMS searches for both region and outcome overlaps.
- 30 4. The GKMS presents the knowledge items retrieved.
5. The expert can then inspect and edit these knowledge items.

Edit Existing Knowledge Items

The knowledge editing process is described above.

35

Inspect History Of Knowledge Items

The expert can inspect the list of knowledge items or mappings in the GKMS with the author status, date defined, and date deactivated (if appropriate) for each item.

5 **Effluent Disposal Advisory System Example**

In the section we consider as example an application related to the discharge of effluents in a river system.

The domain experts first decide the type of application, such as a system for the discharge of effluents in a river. Once the type of applications
10 is decided, the domain experts must specify the problem space and the solution space.

Problem space specification consists of the identification of all the key factors (or attributes) that enter into the definition of an enquiry related to the application field. In our example, some key factors could be: amount and
15 concentration of a restricted substance in the effluent to be discharged and frequency of discharge.

Solution space specification consists of the identification of the components that could be used to define an outcome. In our example, it could include discharge.

20 One can expect that some new factors will have to be added to the problem space and new components to the solution space. Alterations to either the problem or the solution space may need to be done.

An example of a rule might be:

IF ((effluent type is organic substance or pesticide or weedicide)
25 OR (effluent type is pesticide)
 OR (effluent type is weedicide))
AND ((restricted substance is arsenic)
 OR (restricted substance is copper))
AND (0.01 <= concentration <= 0.1)
30 THEN(Carry out the following changes:
 (Neutralise chemical with process y
 And Install concentration monitoring equipment)
 Only then discharge allowed))

Inspection of Figures 10 and 11 show that a significant part of the
35 system relates to the database management. It is therefore natural to use a database management system as part of the development and

implementation tools. The tools used were Microsoft Access and Microsoft Visual Basic. Other tools such as Lotus Notes or Internet technology can be used.

A number of other applications for the technology can be envisaged.

5 For instance: Intelligent (web)Site Advisor (ISA) or Intelligent Resource Advisor (IRA); Intelligent Website Designer or Intelligent Resource Knowledge Designer (IRKD); Product and Service Advisor (PSA); and Intelligent Product and Service Knowledge Designer (PSKD). It should be appreciated that the technology is amenable to network applications.

10 It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to
15 be considered in all respects as illustrative and not restrictive.

CLAIMS:

1. A computerised generic knowledge management system, comprising:
a multi-dimensional global space within computer memory defined by
attributes, where each attribute defines a feature of the external world or the
5 internal state of the system, or actions that can be taken to modify them, and
each attribute is a dimension of the global space;
a source space, within the global space, made up of selected ones of
the attributes to define a context, in which to state problems;
a destination space, within the global space, made of selected ones of
10 the attributes to define a context in which to provide answers to problems
stated in the source space;
mappings between defined parts of the source space which each
represent one or more stated problems, to defined parts of the destination
space which each represent one or more answers expressing and embodying
15 knowledge supplied by experts appropriate to the respective problems stated
in the part of the source space.
2. A system according to claim 1, where the defined parts of the source
and destination spaces are points or regions.
3. A system according to claim 1 or 2, where the destination space is also
20 part of the source space.
4. A system according to claim 1 where the two spaces overlap.
5. A system according to claim 1, where the mapping process are
explanations or actions.
6. A system according to claim 5, where explanation mappings are not
25 calculated, they are stated.
7. A system according to claim 6, where explanation mappings are
associated with code which enable the outcome to be displayed on a screen
or printer.
8. A system according to claim 5, where action mappings associate a
30 situation in the source space to actions expressed in the destination space.
9. A system according to claim 8, where the destination space is made of
instructions to be carried out by agents.
10. A system according to claim 8, where action mappings are specified
by a function or module that can be calculated, using the values of the source
35 attributes that define the situation as parameters.

11. A system according to claim 1, where source and destination space editing sub-systems enables authorised users to define and modify the destination and source spaces or contexts.
12. A system according to claim 1, where a mapping editing sub-system
5 enables experts to define mappings which embody knowledge.
13. A composite system comprising a collection of systems according to claim 1, in which the source contexts of the systems are united, the destination contexts of the systems are united and the mappings of the systems are united to form the composite.
- 10 14. A data acquisition method for a computerised generic knowledge management system, comprising the steps of:
inspecting a problem that either has no answer or an answer which is deemed to be inadequate, that is a problem for which there is no defined part of the source space;
15 specifying attributes, and if appropriate, explanations relevant to the problem;
defining the solution to the problem;
generalising the source context to generalise the inquiry to a larger part of the source space; and
20 saving the knowledge item generated.
15. A method according to claim 14, where the solution is defined after the source context has been generalised.

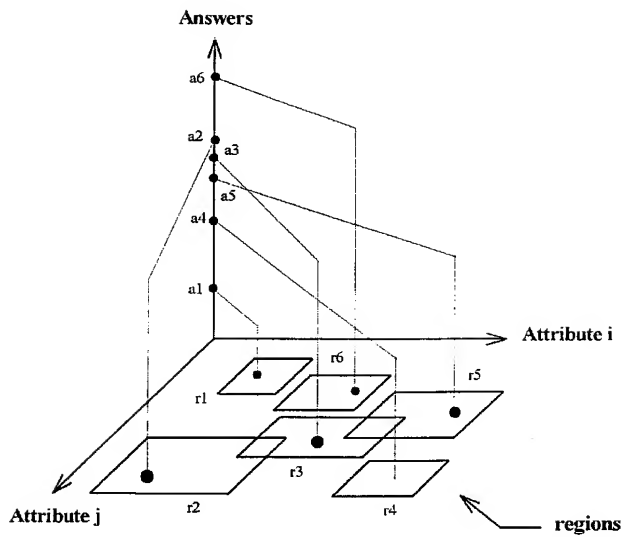
Figure 1

Figure 2

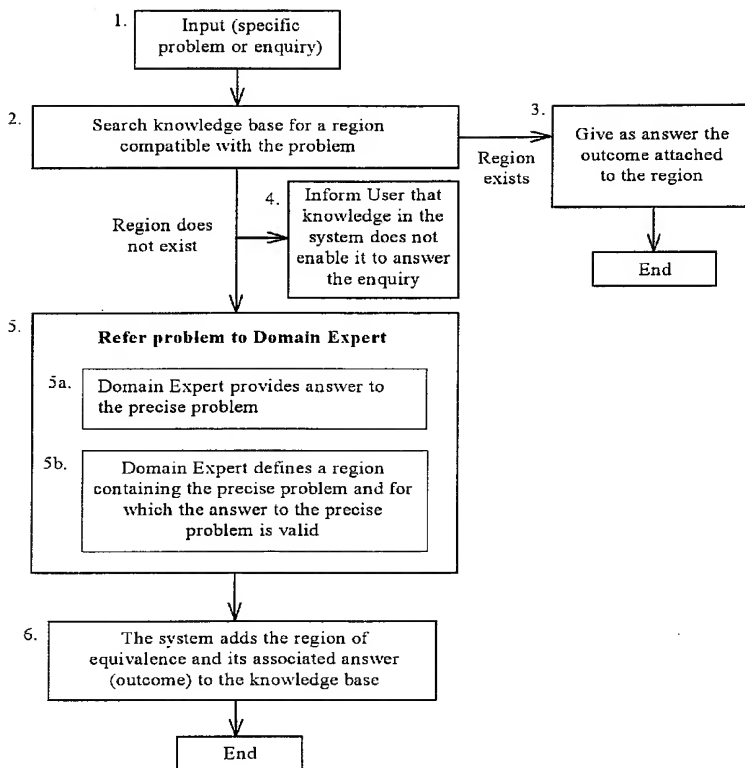


Figure 3

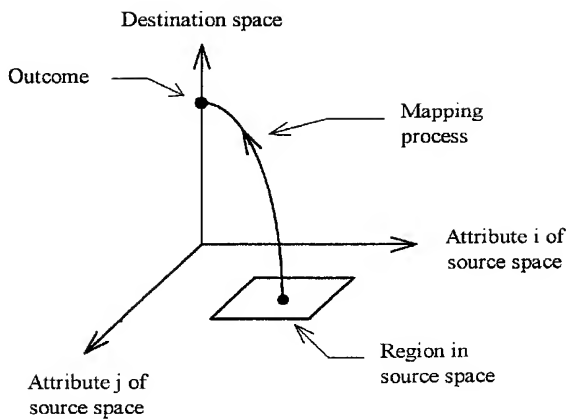


Figure 4

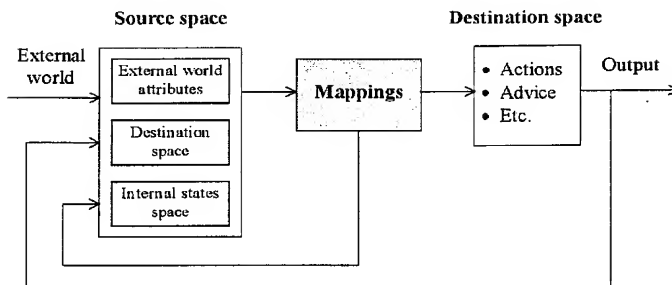


Figure 5

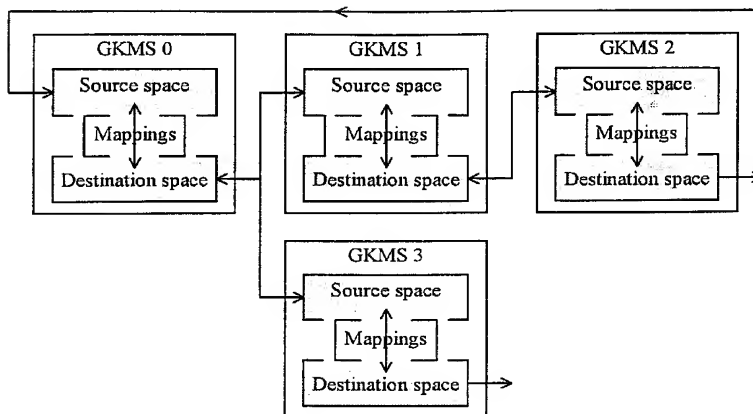


Figure 6

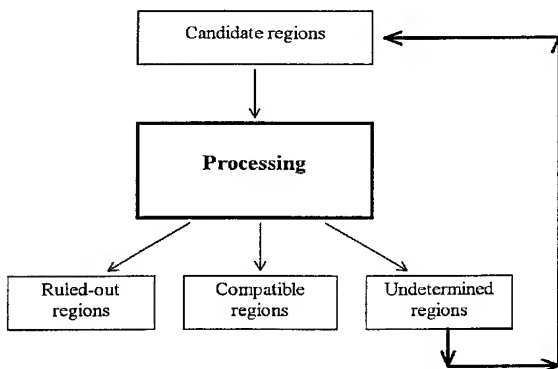


Figure 7

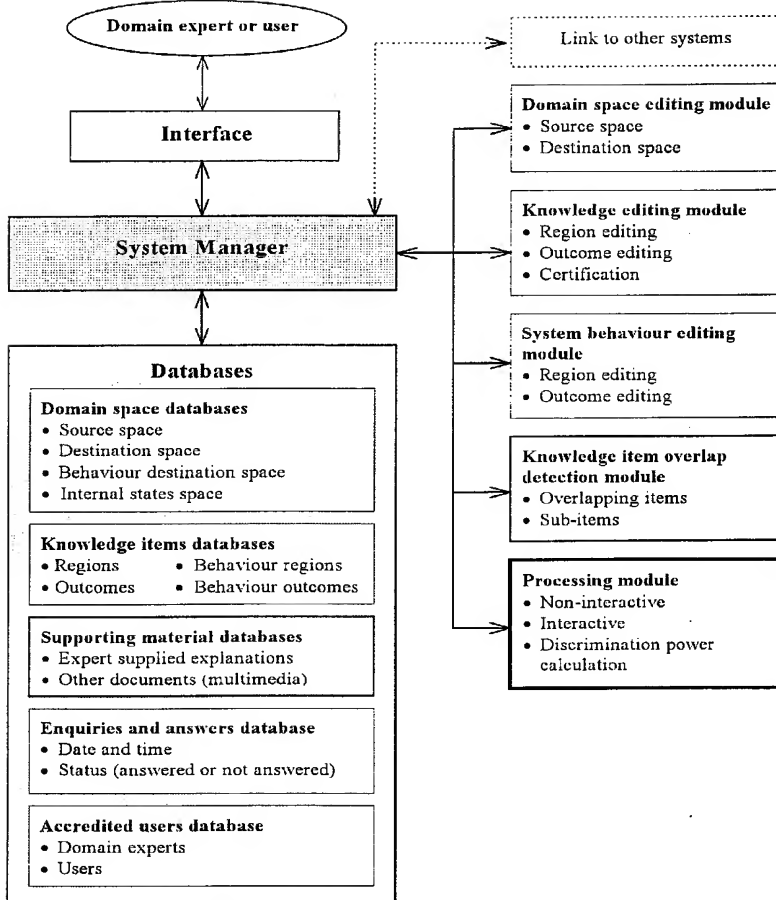


Figure 8

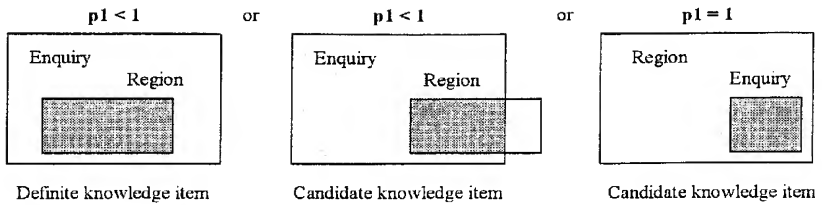


Figure 9

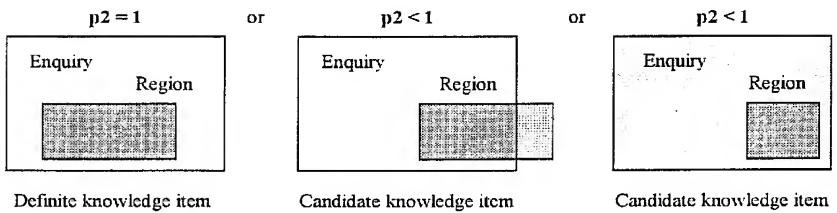


Figure 10

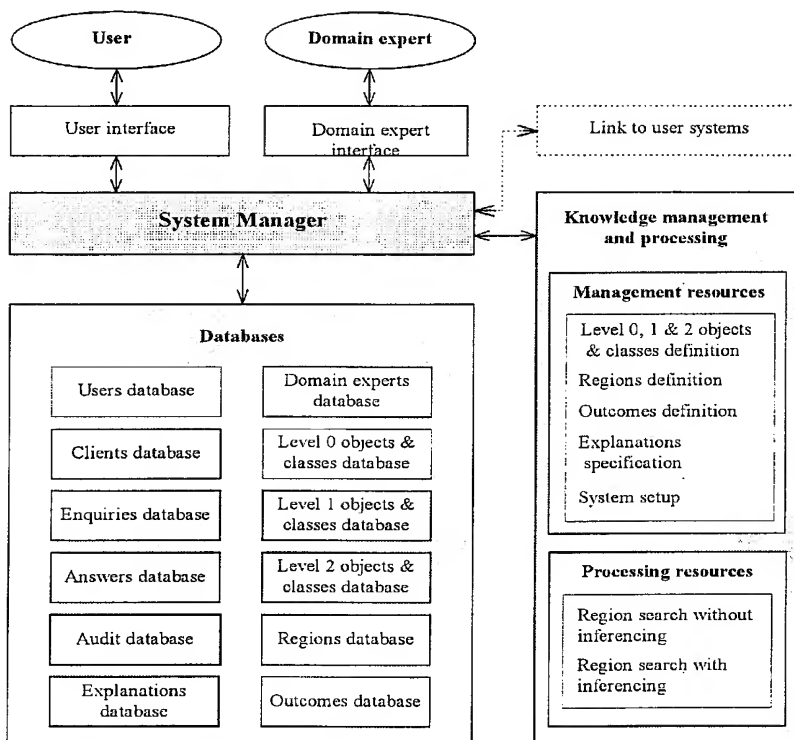
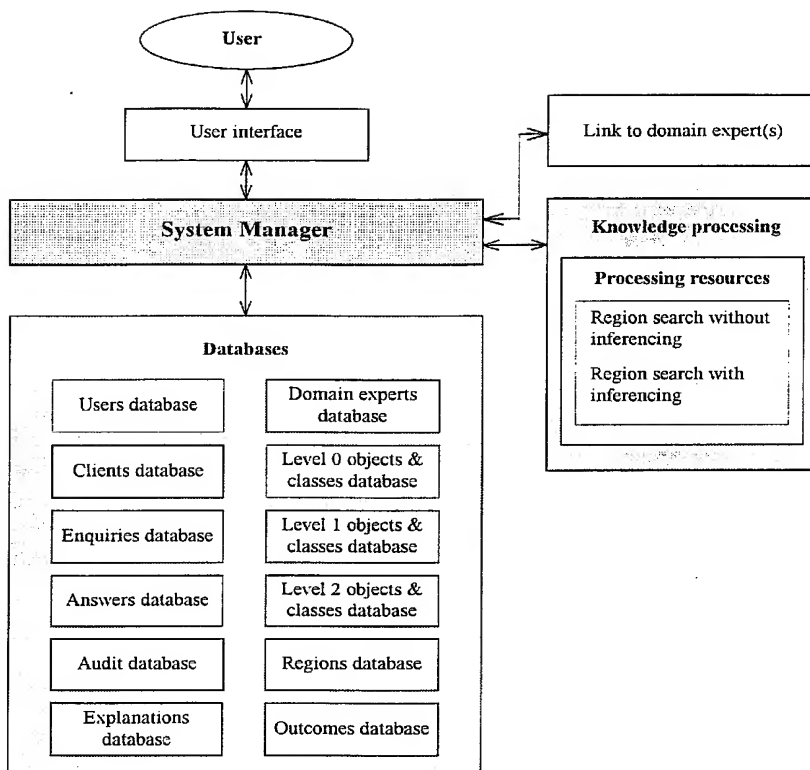


Figure 11



INTERNATIONAL SEARCH REPORT

International application No.

PCT/AU 99/00501

A. CLASSIFICATION OF SUBJECT MATTER		
Int Cl ⁶ : G06F 017/00		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G06F 017/00		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) WPAT, USPTOFT		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A, P	US 5809493 A (Ahamed et al) 15 September 1998 Whole Document	1-15
A	US 5715371 A (Ahamed et al) 3 February 1998 Whole Document	1-15
A	US 5546507 A (Staub) 13 August 1996 Whole Document	1-15
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C <input checked="" type="checkbox"/> See patent family annex		
<p>* Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>		
Date of the actual completion of the international search 6 August 1999		Date of mailing of the international search report 10 AUG 1999
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200 WODEN ACT 2606 AUSTRALIA Facsimile No.: (02) 6285 3929		Authorized officer R.H. STOPFORD Telephone No.: (02) 6283 2177

INTERNATIONAL SEARCH REPORT

International application No.

PCT/AU 99/00501

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5526516 A (Tsuruta et al) 11 June 1996 Whole Document	1-15
A	US 4930071 A (Tou et al) 29 March 1990 Whole Document	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/AU 99/00501

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report				Patent Family Member	
US	5809493				
US	5715371				
US	5546507				
US	5526516	EP	474120	JP	04107635
US	4930071	WO	8810470		

END OF ANNEX